



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/798,474	03/10/2004	Mark Vincent Scardina	50277-2389	7416

42425 7590 11/14/2007  
HICKMAN PALERMO TRUONG & BECKER/ORACLE  
2055 GATEWAY PLACE  
SUITE 550  
SAN JOSE, CA 95110-1089

EXAMINER
----------

TRAN, QUOC A

ART UNIT	PAPER NUMBER
----------	--------------

2176

MAIL DATE	DELIVERY MODE
-----------	---------------

11/14/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

## Office Action Summary

Application No.

10/798,474

Applicant(s)

SCARDINA ET AL.

Examiner

Tran A. Quoc

Art Unit

2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 31 August 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-6, 13-25, 39-44 and 48-69 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-6, 13-25, 39-44 and 48-69 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- ☒ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- ☐ Notice of Informal Patent Application
- ☐ Other: \_\_\_\_\_

### DETAILED ACTION

This is a **Non-Final** Rejection in response to Amendments/Remarks filed 08/31/2007. Claims 1-6, 13-25, 39-44 and 48-69 are pending. Claims 7-12, 26-38, and 45-47 have been cancelled. Claims 1, 13, 39-41 have been amended. Claims 2-6, and 42-44 are previously presented. Claims 14-25 are originally presented, and claims 48-69 are newly added. Effective filing date is 03/10/2004, priority date 09/04/2003 (Oracle).

#### *Interpretations of Claims Language*

It is noted that the terms:

- “*Elements and attributes*” also are read as “*nodes*”, see the current specification at Page 4 Para 13, “while an element is being validated, queries can be issued regarding: (1) the status of the validation, and (2) any processing that may be required with the element in order to conform the element to requirements of an external application. Hence, completely validating the entire XML document or message is not required prior to processing *elements and attributes* (*hereinafter, “nodes”*) within the document. Furthermore, only a single parse of the input stream is necessary. See, disclosure, Page 4 Para 13.

Also, It is noted that the limitation:

- “*while validating a particular XML element in said XML-based input stream, causing said XML processor to generate one or more messages that indicate how to process said particular XML element other than*

Art Unit: 2176

*validating said particular XML element,"*

Upon review of the specification and claims, the Examiner believes Applicants intended this limitation to be defined and function as follows:

In its broadest reasonable interpretation, "***while validating a particular XML element in said XML-based input stream, causing said XML processor to generate one or more messages that indicate how to process said particular XML element other than validating said particular XML element***"

**is also read as:** the XML processor is caused to generate messages that identify the requested annotations in response to the request, and provides for requesting directions regarding such processing and generating messages that identify the processing, or a particular node, while that particular node is being validated without having to wait for validation of the entire XML input prior to receiving requests and generating associated responses, (See the current Specifications Page 13 Para 49 50 and Fig. 1 and Fig. 4).

### ***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

***The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.***

Claim(s) 1-6, 42-44, and 48-69 rejected under 35 U.S.C. 112, second paragraph, as failing to set forth the subject matter which applicant(s) regard as their invention.

Art Unit: 2176

Evidence that independent claims 1, and 48, recite the limitation "**while validating a particular XML element in said XML-based input stream, causing said XML processor to generate one or more messages that indicate how to process said particular XML element other than validating said particular XML element,**" renders the claim indefinite; Since the current specification at Fig. 1, and Page 13 Para 49 merely recites, "XML input stream 104 (FIG. 1). In one embodiment, at block 402, requests for annotations that are associated with elements in an XML-based input stream, are received. At block 404, according to an embodiment, the XML processor is caused to generate messages that identify the requested annotations in response to the request." render the claims indefinite, since one of ordinary skill in the art would not be reasonably appraised of the scope of the invention. See also, MPEP 706.03(d).

Also, claim(s) 1, 48, 50, 53, 54, and 69 rejected under 35 U.S.C. 112, second paragraph, as failing to set forth the subject matter which applicant(s) regard as their invention.

Evidence that claims 1, 48, 50, 53, 54, and 69, recite the limitation "*about said*" renders the claim indefinite, since one of ordinary skill in the art would not be reasonably appraised of the scope of the invention. See also, MPEP 706.03(d).

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

*(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.*

**Claims 1-6, 13-25, 39-44, and 48-69,** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Lovett et al.** US007134072B1 filed 10/13/1999 (hereinafter Lovett), in view of in view of **Fuh et al.** US 20040073870A1 filed 04/16/2003 (hereinafter Fuh).

***Independent claim 1,*** Lovett teaches:

**A method comprising the computer-implemented steps of:**

**while an XML processor performs a validation operation on an XML-based input stream,**

(See Lovett at Column 5, lines 25-60 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser (i.e. validator). The parser 56 parses the received XML data stream, validating the input, and if valid, the XML data is response by forming an in-memory tree representation (i.e. XML DOM).

In addition, Lovett does not explicitly teach, but Fuh teaches:

**while validating a particular XML element in said XML-based input stream, causing said XML processor to generate one or more**

**messages that indicate how to process specific elements in said XML based input stream based on said particular XML element other than validating said particular XML element, by identifying one or more annotations that are associated with said specific elements particular XML element;**

(See Fuh at Page 4 Para 51, discloses the runtime validation engine in accordance with the present invention. XML schema loading module 1010 loads the XML schema in AAE (Annotate Automaton Encoding) format 108, and sets the generic scanner 1017 as the current scanner, via step 1102. Also, an Entity Manager 1008 will obtain the XML external entities referenced by the XML document instance to verify. The current scanner tokenizes the XML document 112. Then, the generic XML parser 1004 calls the current scanner to get a token, via step 1106. The generic XML parser 1004 checks if a token is returned successfully, via step 1108. If the token scan is not successful, then the validation return as "invalid", via step 1110, and the process ends. It is noted the validation notification (i.e. scan process) valid/invalid was generated prior to the validation process. Since, the XML parser and validation module both have to be in the same mode "accept" in order for the validation process to proceed to completion, else the "invalid" will return.

If the token scan is successful, and the generic XML parser 1004 determines that the token is an element or attribute token, via step 1112, then the token is input into the XML schema validation module 1012 as a lexeme, via step 1114. The lexeme can be one of three types: a start tag name, an attribute name,

or an end tag name. Each type of lexeme is processed in a different manner, via steps 1118, 1120, or 1122. If the token is not an element or attribute token, then it is determined if the token is the end of file (EOF) token, i.e., the end of the XML document 112, via step 1124. If not, then the process returns to step 1106 and repeats for the next token. If so, then it is determined if the generic XML parser 1004 and the element validation module 1014 are both in the "accept" mode, via step 1125, i.e., if the parsing and validation has completed. If so, then the validation of the XML document 112 is successful, via step 1126, returning a "valid". If not, then the validation fails, via step 1127, returning an "invalid".

Also, see Fuh at Para 43, discloses element annotation record 829, the first name element annotation record 837, and the notes element annotation record 843.)

**and responding to a request for information about said particular XML element by providing said one or more messages.**

(See Fug 3 Para 43, discloses element annotation record 829, the first name element annotation record 837, and the notes element annotation record 843. (i.e. message).)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Lovett XML validating parser, to include a means of validating a particular XML element in said XML-based input stream, causing said XML processor to generate one or more messages that indicate how to process specific elements in said XML based input stream based on said particular XML element other than validating said particular XML



Art Unit: 2176

element, by identifying one or more annotations that are associated with said specific elements particular XML element and responding to a request for information about said particular XML element by providing said one or more messages as taught by Fuh. One of ordinary skill in the art would have been motivated to perform such a modification, because Lovett and Fuh are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of receiving a request for information about the state of said validation operation; and responding to said request by providing said information about said state of said validation operation, as evidence Lovett at Column 3 Line 50→ Column 4, Line 15, provides a node factory interfaces includes: a namespace, a DTD, a tree builder, and validation node factories to handle, that uses stream of XML tokens, wherein XML-data schemas define items in a way that is order independent. This means that the node factory has to store certain states until it knows it can process those states.)

***Independent claim 13***, Lovett teaches: (As amended)

**A method comprising the computer-implemented steps of:**  
**while performing a validation operation on an XML-based input**  
**stream, performing the steps of: receiving a request for information**  
**about the state of said validation operation; and responding to said**  
**request by providing said information about said state of said**

**validation operation; wherein said information about said state of said validation operation,**

(See Lovett at Column 5, lines 25-60 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser (i.e. validator). The parser 56 parses the received XML data stream, validating the input, and if valid, the XML data is response by forming an in-memory tree representation (i.e. XML DOM).

Also see Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).

**comprises one or more of: the name of a node currently being processed; the data type of the node currently being processed; the current validation mode for the node currently being processed,**

(See Lovett at Fig. 5 and Column 5 Lines 40-60, discloses the validation node factory 68 receives the data elements from the namespace node factory 66.

Also see Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).)

In addition, Lovett does not explicitly teach, but Fuh teaches:

**wherein the current validation mode is one of strict mode, lax mode, and skip mode; and the current state of said validation**

**operation; and annotations that are associated with the node currently being processed.**

(See Fuh at Page 4 Para 51, discloses the generic XML parser 1004 and the element validation module 1014 are both in the "accept" mode, via step 1125, i.e., if the parsing and validation has completed. If so, then the validation of the XML document 112 is successful, via step 1126, returning a "valid". If not, then the validation fails, via step 1127, returning an "invalid". It is noted the Examiner equates the claimed *strict mode, and the current state of said validation operation* as equivalent to the "accept" mode and valid or invalid validation status as taught by Lovett.

Also see Fug 3 Para 43, discloses element annotation record 829, the first name element annotation record 837, and the notes element annotation record 843.)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Lovett's XML validating parser, to include a means of validating the current validation mode is one of strict mode, lax mode, and skip mode; and the current state of said validation operation; and annotations that are associated with the node currently being processed as taught by Fuh. One of ordinary skill in the art would have been motivated to perform such a modification, because Lovett and Fuh are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of receiving a request for information about the state of said validation operation;

Art Unit: 2176

and responding to said request by providing said information about said state of said validation operation, as evidence Lovett at Column 3 Line 50→ Column 4, Line 15, provides a node factory interfaces includes: a namespace, a DTD, a tree builder, and validation node factories to handle, that uses stream of XML tokens, wherein XML-data schemas define items in a way that is order independent. This means that the node factory has to store certain states until it knows it can process those states.)

***Independent claim 39***, Lovett teaches: (As amended)

**A computer-readable medium storing instructions for: a  
validator that validates elements and attributes in an XML-based  
input stream against information that dictates the structure of  
corresponding elements and attributes,**

(See Lovett at Column 5, lines 25-60 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser (i.e. validator). The parser 56 parses the received XML data stream, validating the input, and if valid, the XML data is response by forming an in-memory tree representation (i.e. XML DOM).

Also see Lovett at Fig. 5 and Column 5 Lines 40-60, discloses the validation node factory 68 receives the data elements from the namespace node factory 66.)

**said validator comprising a state machine that responds to requests  
for information about validating a first element in said XML-based  
input stream, while validating said first element;**

Art Unit: 2176

(See Lovett at Column 5, lines 25-60 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser (i.e. validator). The parser 56 parses the received XML data stream, validating the input, and if valid, the XML data is response by forming an in-memory tree representation (i.e. XML DOM).

Also see Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).

Also see Lovett at Fig. 5 and Column 5 Lines 40-60, discloses the validation node factory 68 receives the data elements from the namespace node factory 66.)

**wherein said information about said state of said validation operation,**

(See Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).

**comprises one or more of: the name of a node currently being processed; the data type of the node currently being processed; the current validation mode for the node currently being processed,**

(See Lovett at Fig. 5 and Column 5 Lines 40-60, discloses the validation node factory 68 receives the data elements from the namespace node factory 66.

Also see Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).  
In addition, Lovett does not explicitly teach, but Fuh teaches:

**wherein the current validation mode is one of strict mode, lax mode, and skip mode; and the current state of said validation operation; and annotations that are associated with the node currently being processed.**

(See Fuh at Page 4 Para 51, discloses the generic XML parser 1004 and the element validation module 1014 are both in the "accept" mode, via step 1125, i.e., if the parsing and validation has completed. If so, then the validation of the XML document 112 is successful, via step 1126, returning a "valid". If not, then the validation fails, via step 1127, returning an "invalid". It is noted the Examiner equates the claimed *strict mode, and the current state of said validation operation* as equivalent to the "accept" mode and valid or invalid validation status as taught by Lovett.

Also see Fug 3 Para 43, discloses element annotation record 829, the first name element annotation record 837, and the notes element annotation record 843.)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Lovett's XML validating parser, to include a means of validating the current validation mode is one of strict mode,

Art Unit: 2176

lax mode, and skip mode; and the current state of said validation operation; and annotations that are associated with the node currently being processed as taught by Fuh. One of ordinary skill in the art would have been motivated to perform such a modification, because Lovett and Fuh are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of receiving a request for information about the state of said validation operation; and responding to said request by providing said information about said state of said validation operation, as evidence Lovett at Column 3 Line 50→ Column 4, Line 15, provides a node factory interfaces includes: a namespace, a DTD, a tree builder, and validation node factories to handle, that uses stream of XML tokens, wherein XML-data schemas define items in a way that is order independent. This means that the node factory has to store certain states until it knows it can process those states.)

***Independent Claim 48:*** (New)

is directed to a computer-readable medium to perform the method recited in Claim 1 and is similarly rejected along the same rationale.

***Independent Claim 54:*** (New)

is directed to a computer-readable medium to perform the method recited in Claim 13 and is similarly rejected along the same rationale.

Art Unit: 2176

**Claim 2,**

The rejection of claim 1 is fully incorporated;

in addition Lovett does not explicitly teach, but Fuh teaches:

**receiving a request for said one or more annotations;**

(See Fuh at the Abstract, discloses validating the XML document against the XML schema definition by the XML schema validation parser utilizing the annotated automaton encoding.

Also, see Fug 3 Para 43, discloses element annotation record 829, the first name element annotation record 837, and the notes element annotation record 843 (i.e. message))

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Lovett's XML validating parser, to include a means of receiving a request for said one or more annotations; wherein the step of causing said XML processor to generate one or more messages is performed in response to said request as taught by Fuh. One of ordinary skill in the art would have been motivated to perform such a modification, because Lovett and Fuh are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of receiving a request for information about the state of said validation operation; and responding to said request by providing said information about said state of said validation operation, as evidence Lovett at Column 3 Line 50→ Column 4, Line 15, provides a node factory interfaces includes: a namespace, a DTD, a tree builder, and validation node factories to



Art Unit: 2176

handle, that uses stream of XML tokens, wherein XML-data schemas define items in a way that is order independent. This means that the node factory has to store certain states until it knows it can process those states.)

***Claim 3,***

The rejection of claim 1 is fully incorporated;

in addition Lovett teaches:

**program interface through which information about said  
validation operation can be requested by an external application.**

(See Lovett at Column 5, lines 25-35 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser. The parser 56 parses the XML data stream into individual elements of schema and data, while the data element are validated against the schema and if valid, are used to form an in-memory tree representation of the XML document.

Also, see Lovett at Column 8, Lines 60-65, teaching the Internet (i.e. external environment).)

***Claim 4,***

The rejection of claim 1 is fully incorporated;

in addition Lovett teaches:

**XML processor to generate one or more messages that are  
transmitted in an output stream,**

Art Unit: 2176

(See Lovett at Column 5, lines 25-35 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser. The parser 56 parses the XML data stream into individual elements of schema and data, while the data elements are validated against the schema and if valid, are used to form an in-memory tree representation of the XML document (i.e. output stream).

Also, see Lovett at Column 8, Lines 60-65, teaching the Internet (i.e. external environment).)

***Claim 5,***

The rejection of claim 1 is fully incorporated;

in addition Lovett teaches:

**generate one or more messages before completion of said  
validation operation on said XML-based input stream.**

(See Lovett at Column 5, lines 25-35 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser. The parser 56 parses the XML data stream into individual elements of schema and data, while the data elements are validated against the schema and if valid, are used to form an in-memory tree representation of the XML document (i.e. output stream).

Also, see Lovett at Column 8, Lines 60-65, teaching the Internet (i.e. external environment).)

Art Unit: 2176

**Claim 6,**

the rejection of claim 1 is fully incorporated.

In addition, Lovett does not explicitly teach, but Fur teaches:

**generate one or more messages that indicate how to process said particular XML element, only if said particular XML element is determined valid based on said validation operation on said particular XML element.**

(See Fuh at the Abstract, discloses validating the XML document against the XML schema definition by the XML schema validation parser utilizing the annotated automaton encoding.

Also, see Fug 3 Para 43, discloses element annotation record 829, the first name element annotation record 837, and the notes element annotation record 843 (i.e. message).)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Lovett's XML validating parser, to include a means of generate one or more messages that indicate how to process said particular XML element, only if said particular XML element is determined valid based on said validation operation on said particular XML element as taught by Fuh. One of ordinary skill in the art would have been motivated to perform such a modification, because Lovett and Fuh are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of receiving a request for information about the state of said validation operation; and

Art Unit: 2176

responding to said request by providing said information about said state of said validation operation, as evidence Lovett at Column 3 Line 50→ Column 4, Line 15, provides a node factory interfaces includes: a namespace, a DTD, a tree builder, and validation node factories to handle, that uses stream of XML tokens, wherein XML-data schemas define items in a way that is order independent. This means that the node factory has to store certain states until it knows it can process those states.)

**Claim 14**, Lovett teaches:

**XML-based input stream is defined in corresponding information that dictates the structure of XML data, wherein the step of receiving a request includes receiving a request regarding whether a first element of said,**

(See Lovett at Column 5, lines 25-35 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser. The parser 56 parses the XML data stream into individual elements of schema and data, while the data element are validated against the schema and if valid, are used to form an in-memory tree representation of the XML document.

Also see Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).

**Claim 15**, Lovett teaches:

**wherein the step of receiving a request includes receiving a request regarding what data type definition is associated with first element of said XML-based input stream, wherein said data type is defined in information that dictates the structure of corresponding XML data.**

(See Lovett at the Abstract, discloses the XML document to data type definition (DTD) objects that can be used to validate data elements in the XML document.

Also see Lovett at Column 2 Line 65 → Column 3, Line 40, discloses XML 1.0, data types in the schemas are defined using a set of data type definitions (DTD). The "well-formedness" constraints are those imposed by the definition of XML itself (such as the rules for the use of the < and > characters and the rules for proper nesting of elements). The "validity" constraints are constraints on document structure provided by a particular DTD or XML-Data schema.)

**Claim 16**,

the rejection of claim 15 is fully incorporated.

In addition, Lovett teaches:

**wherein said data type that is associated with said attribute is defined in said information that dictates the structure of corresponding XML data.**

(See Lovett at Column 2 Line 65 → Column 3, Line 40, discloses XML 1.0, data types in the schemas are defined using a set of data type definitions (DTD). The

Art Unit: 2176

"well-formedness" constraints are those imposed by the definition of XML itself (such as the rules for the use of the < and > characters and the rules for proper nesting of elements). The "validity" constraints are constraints on document structure provided by a particular DTD or XML-Data schema.)

***Claim 17,***

the rejection of claim 15-16 are fully incorporated.

In addition, Lovett teaches:

**data type definition in information that dictates the structure of corresponding XML data.**

(See Lovett at Column 2 Line 65 → Column 3, Line 40, discloses XML 1.0, data types in the schemas are defined using a set of data type definitions (DTD). The "well-formedness" constraints are those imposed by the definition of XML itself (such as the rules for the use of the < and > characters and the rules for proper nesting of elements). The "validity" constraints are constraints on document structure provided by a particular DTD or XML-Data schema.)

***Claim 18,***

the rejection of claim 15-16 are fully incorporated.

In addition, Lovett does not explicitly teach, but Fuh teaches:

**a request regarding a first annotation that is associated with first element of said XML-based input stream,**

Art Unit: 2176

(See Fuh at the Abstract, discloses validating the XML document against the XML schema definition by the XML schema validation parser utilizing the annotated automaton encoding.

Also, see Fug 3 Para 43, discloses element annotation record 829, the first name element annotation record 837, and the notes element annotation record 843 (i.e. message))

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Lovett's XML validating parser, to include a means of receiving a request regarding a first annotation that is associated with first element of said XML-based input stream as taught by Fuh. One of ordinary skill in the art would have been motivated to perform such a modification, because Lovett and Fuh are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of receiving a request for information about the state of said validation operation; and responding to said request by providing said information about said state of said validation operation, as evidence Lovett at Column 3 Line 50→ Column 4, Line 15, provides a node factory interfaces includes: a namespace, a DTD, a tree builder, and validation node factories to handle, that uses stream of XML tokens, wherein XML-data schemas define items in a way that is order independent. This means that the node factory has to store certain states until it knows it can process those states.)

Art Unit: 2176

**Claim 19,**

**wherein said information that dictates the structure of corresponding XML data comprises a second definition that is associated with a second element of said XML-based input stream, and wherein the step of receiving a request includes receiving a request regarding said second definition, the method further comprising the computer-implemented step of: before responding to said request regarding said second definition, responding to a request regarding whether said first element is defined in said information that dictates the structure of corresponding XML data.**

(See Lovett at the Abstract, discloses the XML document to data type definition (DTD) objects that can be used to validate data elements in the XML document.

Also see Lovett at Column 2 Line 65 → Column 3, Line 40, discloses XML 1.0, data types in the schemas are defined using a set of data type definitions (DTD). The "well-formedness" constraints are those imposed by the definition of XML itself (such as the rules for the use of the < and > characters and the rules for proper nesting of elements). The "validity" constraints are constraints on document structure provided by a particular DTD or XML-Data schema.)

In addition Lovett does not explicitly teach, but Fuh teaches:

**a second annotation definition that is associated with a second element of said XML-based input stream, and wherein the step of receiving a request includes receiving a request regarding said second annotation, the method further comprising the**



**computer-implemented step of: before responding to said request regarding said second annotation, responding to a request regarding whether said first element is defined in said information that dictates the structure of corresponding XML data.**

(See Fuh at Page 4 Para 51, discloses the runtime validation engine in accordance with the present invention. XML schema loading module 1010 loads the XML schema in AAE (Annotate Automaton Encoding) format 108, and sets the generic scanner 1017 as the current scanner, via step 1102. Also, an Entity Manager 1008 will obtain the XML external entities referenced by the XML document instance to verify. The current scanner tokenizes the XML document 112. Then, the generic XML parser 1004 calls the current scanner to get a token, via step 1106. The generic XML parser 1004 checks if a token is returned successfully, via step 1108. If the token scan is not successful, then the validation return as "invalid", via step 1110, and the process ends. It is noted the validation notification (i.e. scan process) valid/invalid was generated prior to the validation process. Since, the XML parser and validation module both have to be in the same mode "accept" in order for the validation process to proceed to completion, else the "invalid" will return.

If the token scan is successful, and the generic XML parser 1004 determines that the token is an element or attribute token, via step 1112, then the token is input into the XML schema validation module 1012 as a lexeme, via step 1114. The lexeme can be one of three types: a start tag name, an attribute name, or an end tag name. Each type of lexeme is processed in a different manner, via

Art Unit: 2176

steps 1118, 1120, or 1122. If the token is not an element or attribute token, then it is determined if the token is the end of file (EOF) token, i.e., the end of the XML document 112, via step 1124. If not, then the process returns to step 1106 and repeats for the next token. If so, then it is determined if the generic XML parser 1004 and the element validation module 1014 are both in the "accept" mode, via step 1125, i.e., if the parsing and validation has completed. If so, then the validation of the XML document 112 is successful, via step 1126, returning a "valid". If not, then the validation fails, via step 1127, returning an "invalid".

Also, see Fug 3 Para 43, discloses element annotation record 829, the first name element annotation record 837, and the notes element annotation record 843.)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Lovett's XML validating parser, to include a means of receiving a second annotation definition that is associated with a second element of said XML-based input stream, and wherein the step of receiving a request includes receiving a request regarding said second annotation, the method further comprising the computer-implemented step of: before responding to said request regarding said second annotation, responding to a request regarding whether said first element is defined in said information that dictates the structure of corresponding XML data as taught by Fuh. One of ordinary skill in the art would have been motivated to perform such a modification, because Lovett and Fuh are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and

Art Unit: 2176

annotation schema, and provides a predictable result of receiving a request for information about the state of said validation operation; and responding to said request by providing said information about said state of said validation operation, as evidence Lovett at Column 3 Line 50→ Column 4, Line 15, provides a node factory interfaces includes: a namespace, a DTD, a tree builder, and validation node factories to handle, that uses stream of XML tokens, wherein XML-data schemas define items in a way that is order independent. This means that the node factory has to store certain states until it knows it can process those states.)

**Claim 20**, Lovett teaches:

**wherein the step of receiving a request includes receiving a request regarding a status of said validation operation with respect to a first element of said XML-based input stream.**

(See Lovett at Column 5, lines 25-35 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser. The parser 56 parses the XML data stream into individual elements of schema and data, while the data element are validated against the schema and if valid, are used to form an in-memory tree representation of the XML document.

Also see Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).

**Claim 21**, Lovett teaches:

**wherein the step of receiving a request includes receiving a request via an application program interface through which information about said validation operation can be requested by an external application.**

(See Lovett at Column 5, lines 25-35 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser. The parser 56 parses the XML data stream into individual elements of schema and data, while the data element are validated against the schema and if valid, are used to form an in-memory tree representation of the XML document.

Also see Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).

Also, see Lovett at Column 8, Lines 60-65, teaching the Internet (i.e. external environment).)

**Claim 22**, Lovett teaches:

**wherein the step of receiving a request includes receiving a request from an event handler sent in response to an event received in a parser output stream.**

(See Lovett at Column 5, lines 25-35 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser. The parser 56 parses the XML data

stream into individual elements of schema and data, while the data element are validated against the schema and if valid, are used to form an in-memory tree representation of the XML document.

Also see Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).

Also see Lovett at Table 1 Column 6 Line 5-15, teaching the NotifyEvent (i.e. event handler.).

**Claim 23**, Lovett teaches:

**wherein the step of responding to said request includes  
providing, in an output stream, said information about the state of  
said validation operation.**

(See Lovett at Column 5, lines 25-35 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser. The parser 56 parses the XML data stream into individual elements of schema and data, while the data element are validated against the schema and if valid, are used to form an in-memory tree representation of the XML document (i.e. an out stream).

Also see Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).

Art Unit: 2176

**Claim 24**, Lovett does not explicitly teach, but Fuh teaches:

**parsing said XML-based input stream only once for both of  
said validation operation, and operations that are dictated by  
annotations associated with elements in said XML-based input  
stream.**

(See Fuh at Page 4 Para 51, discloses the generic XML parser 1004 and the element validation module 1014 are both in the "accept" mode, via step 1125, i.e., if the parsing and validation has completed. If so, then the validation of the XML document 112 is successful.)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Lovett's XML validating parser, to include a means of parsing said XML-based input stream only once for both of said validation operation, and operations that are dictated by annotations associated with elements in said XML-based input stream as taught by Fuh. One of ordinary skill in the art would have been motivated to perform such a modification, because Lovett and Fuh are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of receiving a request for information about the state of said validation operation; and responding to said request by providing said information about said state of said validation operation, as evidence Lovett at Column 3 Line 50→ Column 4, Line 15, provides a node factory interfaces includes: a namespace, a DTD, a tree builder, and validation node factories to handle, that uses stream of XML tokens, wherein

Art Unit: 2176

XML-data schemas define items in a way that is order independent. This means that the node factory has to store certain states until it knows it can process those states.)

**Claim 25**, Lovett does not teach, but Fuh teaches:

**wherein information that dictates the structure of  
corresponding XML data in said XML-based input stream, with which  
said input stream is validated in said validation operation, comprises  
a plurality of schema definitions that are associated with a plurality  
of corresponding XML documents that could be constituent to said  
XML-based input stream**

(See Fuh at Para 4, discloses the parser receives the XML schema definition and the XML document as input, parses the XML document into a tree format, parses the XML schema definition into a schema tree format, and then traverses the XML document tree to check it against the XML schema tree. The same general-purpose schema validation parser is used for many different XML schemas.

Also see Fuh at Para 7, discloses Each XML schema definition is compiled once into the AAE format, rather than being compiled each time an XML document is validated.)

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified Lovett's XML validating parser, to include a means of said information that dictates the structure of corresponding

Art Unit: 2176

XML data in said XML-based input stream, with which said input stream is validated in said validation operation, comprises a plurality of schema definitions that are associated with a plurality of corresponding XML documents that could be constituent to said XML-based input stream as taught by Fuh. One of ordinary skill in the art would have been motivated to perform such a modification, because Lovett and Fuh are analogous art, since they are from the same field of endeavor of XML parsing and validating input xml data stream and annotation schema, and provides a predictable result of receiving a request for information about the state of said validation operation; and responding to said request by providing said information about said state of said validation operation, as evidence Lovett at Column 3 Line 50→ Column 4, Line 15, provides a node factory interfaces includes: a namespace, a DTD, a tree builder, and validation node factories to handle, that uses stream of XML tokens, wherein XML-data schemas define items in a way that is order independent. This means that the node factory has to store certain states until it knows it can process those states.)

***Claim 40,***

is directed to a computer-readable medium to perform the method recited in Claims 13-15 and is similarly rejected along the same rationale.

***Claim 41:***

is directed to a computer-readable medium to perform the method recited in Claims 13, 22 and is similarly rejected along the same rationale.



**Claim 42:** Lovett teaches:

**reading said annotations from metadata that corresponds to  
said XML-based input stream.**

(See Lovett at the Abstract, discloses processing an Extensible Markup Language (XML) document converts schema elements in the XML document to data type definition (DTD) objects that can be used to validate data elements in the XML document.)

**Claim 43,** Lovett teaches:

**reading said annotations from an XML schema that  
corresponds to said XML-based input stream.**

(See Lovett at the Abstract, discloses processing an Extensible Markup Language (XML) document converts schema elements in the XML document to data type definition (DTD) objects that can be used to validate data elements in the XML document.)

**Claim 44,** Lovett teaches:

**wherein the step of causing said XML processor to generate  
one or more messages includes causing said XML processor to  
generate one or more messages that indicate how to conform said  
specific elements to one or more requirements of an application that  
uses said specific elements.**

(See Lovett at Column 5, lines 25-35 and Table 3 Column 10, Line 50-60, discloses Pull and "push-model" parser. The parser 56 parses the XML data stream into individual elements of schema and data, while the data element are validated against the schema and if valid, are used to form an in-memory tree representation of the XML document.

Also see Lovett at Column 7 Table 2 Lines 10-50, discloses a schema builder interface, where as state machine is reset accordingly, and further determines whether it is valid to have text in the current position within the schema document and generates an error if it is not valid (i.e. error message).

***Claims 49- 53*** respectively: (New)

are directed to a computer-readable storage medium to perform the method recited in Claims 2-6 respectively and are similarly rejected along the same rationale.

***Claims 55-69*** respectively: (New)

are directed to a computer-readable storage medium to perform the method recited in Claims 14-25, 42-44 respectively and are similarly rejected along the same rationale.

It is noted that any citations to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains

Art Unit: 2176

and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. See, MPEP 2123.

### ***Response to Arguments***

The Remarks filed on 08/31/2007 has been fully considered but are moot but in view of the new ground(s) of rejection. This office action is a Non-Final Rejection in order to give the applicant sufficient opportunity to response to the new line of rejection (see rejection for details).

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is 571-272-8664. The examiner can normally be reached on Monday through Friday from 9 AM to 5 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on 571-272-4137. The fax phone number for the organization where this application or proceeding is assigned is (571)-273-8300.

Art Unit: 2176

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Quoc A, Tran/  
Patent Examiner  
Art Unit 2176  
11/08/2007

*/Doug Hutton/*  
Doug Hutton  
Supervisory Primary Examiner  
Technology Center 2100